

OBJECT MODEL MAPPING AND RUNTIME ENGINE FOR EMPLOYING RELATIONAL DATABASE WITH OBJECT ORIENTED SOFTWARE

CROSS REFERENCE TO RELATED APPLICATIONS

A claim of priority is made to U.S. Provisional Patent Application Serial No. 60/069,157, entitled TIER 3 DESIGN SPECIFICATION, filed Dec. 9, 1997 and incorporated herein by reference; and U.S. Provisional Patent Application Serial No. 60/059,939, entitled DATABASE SYSTEM ARCHITECTURE, filed Sep. 26, 1997 and incorporated herein by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

N/A.

BACKGROUND OF THE INVENTION

The present invention is generally related to database technology, and more particularly to interfacing object oriented software applications with relational databases.

The need for interfacing object oriented software applications with relational databases is well known. One method of interfacing an object oriented application with a relational database is to adapt the requests made by the application to the relational database. More particularly, object operations are translated into relational database queries. However, this technique is processor-intensive and sacrifices some of the advantages associated with the object oriented model. As a result, the object oriented software application is unable to function efficiently.

Another method of interfacing an object oriented application with a relational database is to translate database information into a format which is compatible with the object oriented application. Relational databases typically separate data into a plurality of tables through a process known as "normalization" to minimize duplication. A normalized relational database includes a plurality of tables, wherein each table includes at least one field and one key, and at least one field in each table is uniquely dependent upon the key that is associated with the table. These tables can be translated into objects. However, the objects can become inaccurate when changes are made to the relational database. It is known to adapt to changes in the relational database by performing further translations, but this process requires substantial effort.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a mapping between an object model and a relational database and a runtime engine are employed to facilitate access to a relational database. The object model can be created from database schema or database schema can be created from the object model. Further, the mapping can be automatically generated. The database schema, object model, and mapping are employed to provide interface objects that are utilized by an object oriented software application to access the relational database.

The present invention provides transparent access to the relational database. The interface objects and runtime engine perform read and write operations on the database, including generation of SQL code. Consequently, neither programmers nor software applications need have knowledge of the

database structure, the database programming interface, database security, or the database transaction model in order to obtain access to the relational database. Further, changes to the relational database do not always necessitate additional mapping.

BRIEF DESCRIPTION OF THE DRAWING

Other features and advantages of the present invention will become apparent in light of the following detailed description of the drawing, in conjunction with the drawing, of which:

FIG. 1 is a block diagram that illustrates use of the map to generate interface objects that are employed by a runtime engine and an object oriented software application to access a relational database;

FIG. 2 is a block diagram of database schema;

FIG. 3 is a block diagram of an object model;

FIG. 4 is an object diagram of a mapping;

FIG. 5 is an object diagram of the runtime engine;

FIG. 6 is an object diagram of RtCore.DLL; and

FIG. 7 is a sequence diagram that illustrates operation of the runtime engine.

DETAILED DESCRIPTION OF THE INVENTION

Referring to FIG. 1, a mapping tool 10 is employed to generate a map 12 in which relationships between an object model 14 and schema associated with a relational database 16 are defined. A code generator 18 is employed to examine the relationships that are defined in the map 12 and a model object oriented interface associated with an object oriented software application 22 to generate interface objects 20. The interface objects 20 are employed by the object oriented software application 22 to access the relational database 16 via a runtime engine 24, which also uses the map 12 to drive its processing.

The object model 14 is a template that has a predetermined standardized structure. The illustrated object model includes attributes and inheritance relationships that are mapped to relational database features such as tables, rows, columns, keys, and foreign keys. Mapping the object model to the relational database schema includes mapping a class attribute to a table column, mapping a class attribute to a 1-1, 1-N, or N-N relationship, and mapping class inheritance to rows within a table or across tables.

Referring now to FIGS. 2, 3 and 4, the mapping of a class attribute to a table column can be described generally as: Class Attribute→Table Column+Class Key+Joins. Mapping the class attribute defines where the attributes are read from and written to. In the illustrated example, the class attribute CPerson.name 26 maps to table column TPerson.name 28. The "Class Key" is employed to relate an object instance to a row in the table. In particular, key values of the class are mapped to columns in a table that comprise the primary key. In the illustrated example, CPerson.id 30 maps to TPerson.id 32. "Joins" defines keys between tables within a class. Since there is only one table in the mapping of Cperson.name to Tperson.name, no information is required for Joins. If Cperson includes two tables, such as Tperson and X, then mapping Cperson.y to X.y includes: Cperson.y Maps to X.y+Cperson.id Keys to Tperson.id+Tperson.id Joins to X.id.

Mapping a class attribute to a 1-1, 1-N, or N-N relationship with at least one other object can be described generally